IN THE CLAIMS

Please amend the claims as follows.

1. (Original) A system comprising:

a memory;

a plurality of pages held in the memory;

an instruction translation look aside buffer (ITLB);

a first data translation look aside buffer (DTLB);

a translation look aside (TLB) miss handler; and

an executable/non-executable (x) indicator associated with each page in memory wherein the TLB miss handler sets the x-indicator for a particular page to indicate "non-executable" when that page is accessed in a mode that allows writing to that page, and wherein the ITLB refuses to allow instructions from a page with an associated x-indicator of "non-executable" to be loaded.

2. (Original) The system of claim 1, wherein the TLB miss handler is implemented at least in part as software.

3. (Original) The system of claim 1, wherein the DTLB is utilized only for vector accesses to memory.

4. (Original) The system of claim 1, wherein the DTLB is utilized only for scalar accesses to memory.

5. (Original) The system of claim 1, further comprising for each page in memory:

a read bit indicating that the page is valid and readable; and

a write bit indicating that the page is valid and writable.

6. (Original) The system of claim 1, further comprising a write bit associated with each page in memory that indicates the respective page is writable.

7. (Original) The system of claim 1, further comprising a page table used to translate a virtual address to a real address, wherein the x-indicator for each page is held in the page table entry associated with that page.

8. (Original) The system of claim 7, wherein the page table used to translate a virtual address to a real address includes a write bit in each page table entry.

9. (Original) The system of claim 7, wherein the page table used to translate a virtual address to a real address includes:

a write bit in each page table entry; and

a read bit in each page table entry.

10. (Original) A system comprising:

a memory;

a plurality of pages held in the memory;

an instruction translation look aside buffer (ITLB);

an ITLB miss handler associated with the ITLB;

a data translation look aside buffer (DTLB);

a DTLB miss handler associated with the DTLB; and

an executable/non-executable x-indicator associated with each page in memory wherein the DTLB miss handler sets the x-indicator for a particular page to indicate "non-executable" when that page is accessed in a mode that allows writing to that page, and wherein the ITLB refuses to load instructions from a page if the x-indicator indicates a "non-executable" state.

11. (Original) The system of claim 10, wherein the ITLB miss handler sets the x-indicator to an "executable" state when a page is valid and executable.

12. (Original) The system of claim 10, wherein the ITLB miss handler is implemented at least in part as software.

13. (Original) The system of claim 10, wherein the DTLB miss handler is implemented at least in part as software.

14. (Original) The system of claim 10, wherein the DTLB is utilized only for scalar accesses to memory.

15. (Original) The system of claim 10, wherein the DTLB is utilized only for vector accesses to memory.

16. (Original) The system of claim 10, further comprising for each page in memory:

a read bit indicating that the page is valid and readable; and

a write bit indicating that the page is valid and writable.

17. (Original) The system of claim 10, further comprising a page table used to translate a virtual address to a real address, wherein the x-indicator for each page is held in the page table entry associated with that page.

18. (Original) The system of claim 17, wherein the page table used to translate a virtual address to a real address includes a write bit in each page table entry.

19. (Original) The system of claim 17, wherein the page table used to translate a virtual address to a real address includes:

a write bit in each page table entry; and

a read bit in each page table entry.

20. (Original) The system of claim 10, further comprising a write bit associated with each page in memory that indicates the respective page is writable.

21. (Currently amended) A computerized method comprising:

providing a system with a processor having a memory;

holding a plurality of pages in the memory;

translating an ~~addresses~~ address for an instruction fetch; and

translating an ~~addresses~~ address for a data access, wherein the translating of the address for the instruction fetch includes determining whether the address has an executable indication associated with the address and only if executable then continuing with the instruction fetch, and is done differently than the translating of the ~~addresses~~ address for the data access, and

wherein the translating of the address for the data access includes setting a ~~non~~ non-executable indication.

22. (Original) The method of claim 21, wherein setting of a non executable indication is done for a page holding the data for a data access.

23.-24. (Cancelled)

25. (Currently amended) The method of claim 21, wherein the translating of the address for the data access also includes setting the non executable indication to "on" for a page holding the data access address ~~on~~ if a write indication is set for that page.

26. (Original) The method of claim 21, wherein the non executable indication is set for a page holding the data access address in response to finding a write indication to allow writing to the page.

27. (Currently amended) A computer program product for use with a computer system having a processor and memory, the computer program product associated with translating virtual addresses to real addresses, the computer program product comprising a ~~computer~~ computer-usable medium having a set of instructions executable by a suitably programmed information handling system embodied in the computer usable medium for causing the computer system to execute a method comprising:

holding a plurality of pages in the memory;

translating an ~~addresses~~ address for an instruction fetch; and

translating an ~~addresses~~ address for a data access, wherein the translating of the address for the instruction fetch includes determining whether the address has an executable indication associated with the address and only if executable then continuing with the instruction fetch, and is done differently than the translating of the ~~addresses~~ address for the data access, and

wherein the translating of the address for the data access includes setting a non executable indication.

28.-29. (Cancelled)

30. (Original) The computer program product of claim 27, further comprising instructions wherein the translating of the address for the data access also includes setting the non executable indication for a page holding the data access address on if a write indication is set for that page.

31. (Original) The computer program product of claim 27, further comprising instructions wherein the non executable indication is set for a page holding the data access address in response to finding a write indication to allow writing to the page.

32. (Original) An apparatus for translating virtual to real addresses in a computing system having a processor and a memory, the apparatus comprising:

means for holding a plurality of pages in the memory;

means for translating addresses for instructions;

means for translating addresses for data, wherein means for translating addresses for instructions operates separately from means for translating addresses for data, and

wherein means for translating the address of a page for a data access further includes means for setting a non executable indication.

33. (Original) The apparatus of claim 32, wherein means for setting the non executable indication is done for a page holding the data for a data access.

34. (Currently amended) The apparatus of claim 32, wherein means for translating the addresses of ~~instruction fetch~~ the instructions includes means for checking the non executable indication.

35. (Currently amended) The apparatus of claim 32, wherein means for translating ~~of~~ the ~~address for~~ addresses of the ~~instruction fetch~~ instructions includes:

means for checking the non executable indication; and

means for disallowing an instruction-buffer load based on ~~fetch in response to finding~~ the non executable setting.

36. (Currently amended) The apparatus of claim 32, wherein means for translating of the address for the data access also includes means for setting the non executable indication to "on" for a page holding the data for the data access address ~~on~~ if a write indication is set for that page.

37. (New) A system comprising:

a memory;

a plurality of pages held in the memory;

an instruction buffer;

a translation look aside buffer (TLB) having a plurality of TLB entries, each of the plurality of TLB entries having an instruction/data (I/D) indicator and data from a page in memory;

a translation look aside (TLB) miss handler; and

an executable/non-executable (x) indicator associated with each page in memory, wherein the TLB miss handler sets the I/D-indicator to be loaded into a particular TLB entry to indicate "instructions" when the associated page has its associated x-bit indicating "executable" and to indicate "data" when the associated page has its associated x-bit indicating "non-executable," and wherein the system refuses to allow data from a page indicated as "data" to be loaded into the instruction buffer.

38. (New) The apparatus of claim 37, further comprising:

a cache having a plurality of cache lines, each cache line having an instruction/data (I/D) indicator, wherein, based on a TLB miss, the TLB miss handler loads at least one cache line with data from a page in memory and with an I/D indication based on the x-indicator associated with that page in memory.

39. (New) The system of claim 37, wherein the DTLB includes a first portion used only for vector accesses to memory.

40. (New) The system of claim 37, wherein the DTLB includes a second portion used only for scalar accesses to memory.

41. (New) A system comprising:

a memory;

a plurality of pages held in the memory;

an instruction translation look aside buffer (ITLB);

a first data translation look aside buffer (first DTLB);

a translation look aside (TLB) miss handler; and

an executable/non-executable (x) indicator associated with each page in memory wherein the x-indicator for a particular page is set to indicate "non-executable" when that page is accessed in a mode that allows writing to that page, and wherein the TLB handler refuses to load the ITLB from a page with an associated x-indicator of "non-executable".

42. (New) The apparatus of claim 41, further comprising:

a cache having a plurality of cache lines, each cache line having an instruction/data (I/D) indicator, wherein, based on a TLB miss, the TLB miss handler loads at least one cache line with data from a page in memory and with an I/D indication based on the x-indicator associated with that page in memory.

43. (New) The system of claim 41, wherein the first DTLB is used only for vector accesses to memory.

44. (New) The system of claim 41, further comprising a second DTLB, wherein the second DTLB is used only for scalar accesses to memory.